An Intelligent Decision Support System for Crane Scheduling in a Container Terminal

Guohua WAN

Faculty of Business Administration, University of Macau, Taipa, Macao SAR, China

E-mail: ghwan@umac.mo; Telephone: +853-3974751; Fax: +853-838320

Abstract In this paper, we studied a crane scheduling problem, i.e., the scheduling of RMCs to minimize the total weighted remaining work for the blocks in a terminal storage yard. The approach is a mix of planning method from AI and optimization methods based on MIP formulation. We implemented a DSS to incorporate both AI techniques and OR models. The experiments in simulation environment showed promising scheduling results with the DSS prototype.

Keywords Crane Scheduling; Decision Support System; Planning; Mixed Integer Programming

1. Introduction

Today, sea transportation has been playing a vital role in promoting international trade. As a part of sea transportation systems, container terminals serve as essential hubs in the sea transportation as well as overall transportation network. Basically, container terminals fulfill three tasks [9]:

- Delivering containers to and receiving them from shippers and consignees;
- Loading and discharging containers to and from vessels; and
- Temporarily storing containers.

In a container terminal system, these basic tasks include: (1) berth allocation, (2) yard planning, (3) stowage planning, (4) crane (including quay cranes and yard cranes) scheduling, and (5) logistics planning of operations.

Logistics planning deals with assigning and coordinating the operations of port equipments (yard space, quay cranes, yard cranes and prime movers). It decides what parts of a container yard are going to be assigned to containers, and when and what cranes will be scheduled to move the containers. In a container terminal it is a critical task because usually yard space and cranes are scarce resources and the planning of those facilities has strong impact on the overall performance of the container terminal system.
In this paper, we consider a crane-scheduling problem in a container terminal. Crane is one of most frequently used equipments for container handling in container terminals and most cranes are huge equipments, often installed on rails, track or tires, and difficult to be controlled and scheduled [1][5][8]. Making tradeoff between the complexity of scheduling and efficiency, human beings, in practice, often sacrifice efficiency to get easy scheduling solutions. However, with the emergence and development of the scale of economy, it is easy to win a great deal of cost saving with a bit improvement of scheduling solutions. Conversely, a poor and inefficient scheduling will lead to a higher cost and bottleneck. The pressure for more efficient crane scheduling is increasing in the global and highly competitive economy.

2. Problem description

According to the manner of crane movement, basically, there are three types of cranes: the cranes mounted on a tower or a fixed point, could not move at all, and most of cranes used in the construction industry belong to this type; the cranes installed on a truck or tires, could move flexibly; the cranes mounted on rails or tracks, where the crane movements are restricted only on the tracks or rails.

For simplicity, we call the crane mounted on rails or tracks as rail-mounted crane (RMC). RMC is prevalent in the transportation industry, for instance, rail-mounted gantry yard cranes in the container terminals. In fact, crane scheduling is similar to the job scheduling in the sense that cranes are regarded as machines, and the tasks of cranes as jobs. The problem is to assign jobs to cranes so that some objective function is maximized or minimized. But for RMC, there is big difference. RMC run on a track or rail, the spatial constraint for cranes not to cross must be respected if several cranes are installed on the same track or rail. The spatial-constraint complicates the crane scheduling and makes the problem more difficult to solve than the job scheduling.

In a container terminal, the container storage yard is used to temporarily store the inbound and outbound containers for being picked up by trucks or loaded onto vessels. A large-scale yard can be divided into a number of large areas called zones. In each zone, containers are stacked side by side and one on top of the other to form rectangular shaped heaps called blocks. A typical block may have 6 lanes of containers placed side by side, 5 containers in height for each lane and usually more than 20 containers in length, depending on the geographical shape of the storage yard. Container retrieval and stacking within each block are handled by RMCs [12].

Due to vessel schedules and the policies for container allocation in the yard, the workload of blocks varies with time. In order to finish the work within the planned time so that vessels
can leave at the prescribed departure times, it is important to determine an efficient schedule for RMCs since RMCs are very expensive equipments, and terminals usually cannot allocate and keep a fixed number of RMCs for each block. To effectively and efficiently utilize the RMCs and to avoid the imbalance of workload among blocks, an RMC might move from a block to another block. The allocation and the movement of RMCs among blocks are collectively called crane scheduling. Since RMCs are big in size and slow in motion, their movements occupy a large amount of road space for a long time, obstructing the traffic on the rails between blocks and delaying other terminal operations. Furthermore, a block-to-block RMC movement takes long time, thus results in the loss of the productivity of RMCs. Therefore, determining the optimal RMC schedule among blocks is critical to the efficiency of container handling operations in the storage yard [12].

The container yard operates several shifts around the clock every day. Every evening, the data of workload for the following day is provided. The schedule of RMCs is determined for each of the planning periods, by crane operation supervisors, usually based on experience.

Basically, given the data of workload in each block of a planning period, the schedulers make the following two decisions so as to minimize the total cost of RMCs within a period:

1. When to move an RMC and,
2. How to move an RMC (i.e., the origin and the destination blocks for an RMC).

While the schedulers have good knowledge and experience for scheduling RMCs, the large number of RMCs and blocks practically precludes human beings from consistently coming out sound schedules. In the following, we propose a hybrid approach to solve the problem.

3. An approach integrating artificial intelligence and optimization to crane scheduling

Although optimization methods play important roles in solving scheduling problems, their effectiveness and efficiency are yet to be fully realized in transportation and logistics industry with large volume of data, lots of dimension of variables and complex settings [4][7].

In order to solve the problem efficiently and effectively, we propose a hybrid approach that consists of two stages. First, planning method from artificial intelligence (AI) is employed to decompose the scheduling problem into planning task (decide which RMCs and container zones are operated) and scheduling task (decide when the operations will be performed). Second, the optimization methods are studied to solve the RMCs scheduling problems. The optimization method is based on mixed integer programming formulation, and is solved using Lagrangian relaxation method.
3.1 Planning for RMCs scheduling

In this stage, the scheduling problem is decomposed into a set of planning tasks using AI techniques [10]. The purpose is to decompose the whole complicated problem into a set of sub-problems, which involves only part of the blocks and the RMCs in the terminal. The sub-problem is to schedule the RMCs within some blocks. The planning method can be represented in situation calculus by logical sentences as follows:

- **Initial state**: it is a logical sentence about a situation $S_0$. For our problem, this might be a conjunction of a list of $At(Block(i), RMC(j))$, which means that $RMC(j)$ is at $Block(i)$.
- **Goal state**: it is a logical query asking for suitable situations. For our problem, this might be a conjunction of a list of $At(Block(k), RMC(l))$, which requests $RMC(l)$ at $Block(k)$.
- **Operators**: they are a set of descriptions of actions using the action representation. For example, the following is a successor-state axiom involving moving a RMC from block $k$ to another adjacent block:

  \[
  \forall a, s: \text{At}(s, RMC(j)) \Leftrightarrow [a=\text{Move}(RMC(j)) \land \text{At}(Block(k), RMC(j)) \\
  \land \text{Adjacent}(s, Block(k))] \\
  \lor [\text{At}(s, RMC(j)) \land a \neq \text{Move}(RMC(j))]
  \]

Here, in the planning methods, we use STRIPS language to represent states and goals. Both the states and goals are represented by conjunctions of function-free ground literals, i.e., predicates applied to constant symbols, possibly negated. The representation of initial states and goal are used as input to the planning system. The operators in STRIPS language consist of three components. The action description is what the planning system actually returns in order to do something. Within the planner it serves only as a name for a possible action. The precondition is a conjunction of atoms (positive literals) that says what must be true before the operator can be applied. The effect of an operator is a conjunction of literals (positive or negative) that describes how the situation changes when the operator is applied. For instance, a STRIPS operator for moving from RMCs from one block to another block is as follows:

\[
\text{Op}(Action: \text{Move}(Block(i), RMC(j)), Precondition: \text{At}(Block(k), RMC(j)) \\
\land \text{Path}(Block(i), Block(k)), Effect: \text{At}(Block(k), RMC(j)) \land \neg \text{At}(Block(i), RMC(j)))
\]

A plan is formally defined as a data structure consisting of the following four components:

- A set of steps. Each step is one of the operators for the problem.
- A set of step ordering constraints. Each ordering constraints is of the form $S_i \subset S_j$, which means “$S_i$ before $S_j$” thus step $S_i$ must occur sometime before step $S_j$ (but not necessarily immediately before).
A set of variables binding constraints. Each variable is of the form \(v=x\), where \(v\) is variable in some step, and \(x\) is either a constant or another variable.

A set of causal links. A causal link is of form \(S_i \rightarrow^c S_j\), which means \(S_i\) achieves \(c\) for \(S_j\). Causal links serve to record the purpose of steps in the plan, meaning that a purpose of \(S_i\) is to achieve the precondition \(c\) of \(S_j\).

The initial plan simply describes the problem to be solved. It consists of two steps, \(Start\) and \(Finish\) with the ordering \(Start \subseteq Finish\). Both \(Start\) and \(Finish\) have null actions associated with them, thus when it is time to execute them they are ignored. The \(Start\) step has no preconditions, and its effect is to add all the propositions that are true in the initial state. The \(Finish\) step has the goal state as its precondition, and no effects. By defining the scheduling problem in this way, the planning system can start with the initial plan and manipulate it until they come up with a plan that is a solution. The plan for our RMC scheduling problem can be sketched as follows:

\[
\text{Plan}(\text{STEPS}: \{ S_1: \text{Op(ACTION: Start)}, S_2: \text{Op(ACTION: Finish)}, \text{PRECONDITION: ...} \}).
\text{ORDERING:} \{ S_1 \subseteq S_2 \},
\text{BINDING:} \{ \},
\text{LINKS:} \{ \}
\]

A solution is a plan that guarantees achievement of the goal, i.e., a solution is a complete and consistent plan [10].

Using the approach described above, we can come out a plan, which indicates what RMCs should move to what blocks (in a zone) with the path for movement. Then, based on this plan, we can use optimization method to derive a schedule for each of the zones.

3.2 Scheduling of RMCs using optimization methods

In this stage, the sub-problem of RMCs scheduling is formulated as mixed integer programming, and is solved using Lagrangian relaxation method.

First, we assume in our RMC scheduling model, the following statements are true.

- The capacity of an RMC is measured in minutes. All RMCs have the same capacity in each planning period. Similarly, the workload is also measured in minutes. The total number of containers to be discharged from and loaded onto vessels during each planning period is known in advance.
- Because of the limitation of block sizes and the potential danger of crane collision, each
block holds at most two RMCs at any time. With at most only two RMCs per block, it is not allowed more than one RTGC to move from one block to another in one period.

- Also, no inter-zone RMC movements are carried out after planning.
- Any crane move starts and finishes within the same planning period. Unfinished work in a block at the end of a planning period will be carried over to the next period. As a result, the workload of a block in a planning period is the sum of the workload of the current period and the workload carried over from the previous planning period. The workload carried over from the previous period will be finished during the early part in the current period. The unfinished workload in a period is called “delayed work”.

The scheduling problem can be formulated as a MIP model to find the optimal movements of RMCs and the time of the movements in a zone during each planning period, to minimize total weighted remaining of work, which might be discharging from or loading onto vessels during each planning period. Now we define the following notations for our model (we use notations similar to what used in [12]):

- \( X_{ij0} \): the number of cranes in block \( i \) at the beginning of the planning horizon,
- \( P \): the capacity of an RMC within a planning period;
- \( N \): the total number of blocks in a zone;
- \( T \): the total number of planning periods in a planning horizon;
- \( B_{it} \): the workload in block \( i \) within planning period \( t \),
- \( t_{ij} \): the traveling time for an RMC moving from block \( i \) to block \( j \).
- \( w_{it} \): the weight of work in block \( i \) at the end of planning period \( t \).
- \( X_{ijt} \): the number of RMCs moving from block \( i \) to block \( j \) during planning period \( t \) (note that if \( i=j \), these RMCs stay in the same block during period \( t \));
- \( Z_{ijt} \): the workload fulfilled in block \( i \) by cranes that move from block \( i \) to block \( j \) during planning period \( t \);
- \( Y_{ijt} \): the workload fulfilled in block \( j \) by cranes that move from block \( i \) to block \( j \) during planning period \( t \);
- \( R_{it} \): the remaining workload in block \( i \) at the end of planning period \( t \).

Then, we have the following model:

\[
\begin{align*}
\text{Min} \ J = & \sum_{t=1}^{T} \sum_{i=1}^{N} w_{it} R_{it}^2 \\
\text{Subject to:}
\sum_{j=1}^{N} X_{ij} = & \sum_{j=1}^{N} X_{j(i-1)} \quad \text{for } i=1, 2, \ldots, N, \ t=1, 2, \ldots, T \ (1) \\
\sum_{j=1}^{N} X_{ij} + \sum_{j=1, j \neq i}^{N} X_{j(i-1)} \leq 2 \quad \text{for } i=1, 2, \ldots, N, \ t=1, 2, \ldots, T \ (2) \\
R_{it-1} + B_{it} + \sum_{j=1}^{N} Z_{ijt} + \sum_{j=1}^{N} Y_{ijt} = & \ R_{it} \quad \text{for } i=1, 2, \ldots, N, \ t=1, 2, \ldots, T \ (3) \\
Z_{ijt} + Y_{ijt} \leq & \ (P - t_{ij}) X_{ijt} \quad \text{for } i, j=1, 2, \ldots, N, \ t=1, 2, \ldots, T \ (4) \\
R_{0i} = & \ 0 \quad \text{for } i=1, 2, \ldots, N \ (5)
\end{align*}
\]
\[ X_{ij} = 0 \quad \text{for } i, j = 1, 2, \ldots, N, i \neq j \quad (6) \]
\[ R_{it} \geq 0, Z_{ijt} \geq 0, Y_{ijt} \geq 0 \quad \text{for } i, j = 1, 2, \ldots, N, t = 1, 2, \ldots, T \quad (7) \]
\[ X_{iit} \in \{0, 1, 2\}, X_{ijt} \in \{0, 1\} \quad \text{for } i, j = 1, 2, \ldots, N, \text{ and } i \neq j, t = 1, 2, \ldots, T \quad (8) \]

Note that the remaining workload in a block is some sort related to tardiness, which is an important performance measure of scheduling solution. The above constraints are easy to understand, so we just do not give more explanation here.

The model can be solved using standard Lagrangian relaxation, see [3] [6] [12] for details.

4. The design of a decision support system and experiments

Based on the approach in last section, we developed a decision support system for RMC scheduling in a container terminal. It consists of four parts: a problem solver, which contains an artificial intelligence based planning module and an MIP (mixed integer programming) model to help solve the scheduling problem; a database which stores relevant data (positions and number of zones, position and capacity of RMCs, and requirements of the operations—constraints, etc.); and a graphical user interface, through which the user can adjust the schedule produced by the system manually, which is very important in real operations [2][11].

The procedure for the DSS to solve a RMC scheduling problem is sketched in the following.

\[
\text{BEGIN}
\]
\begin{itemize}
\item \textbf{Initialize} the positions of the RMCs and workload of each zone
\item \textbf{Call} the planning module to allocate the RMCs to zones,
\item \hspace{1cm} to balance the workload of each zone
\item \textbf{FOR} each zone:
\item \hspace{1cm} \textbf{Call} the optimization module to schedule the RMCs in that zone
\end{itemize}
\[
\text{ENDWHILE}
\]
\[
\text{END}
\]

In the first stage, the proposed approach is experimented with several data sets in a simulation environment with a decision support system prototype. The results show that the approach is effective and efficient to produce promising solutions to the difficult scheduling problem. The decision support system will be further extended for use in a real container terminal system.

5. Conclusions

In this paper, we studied an important scheduling problem in container terminals, i.e., the scheduling of RMCs to minimize the total weighted remaining work for the blocks in a
terminal storage yard. The approach is a mix of planning method from AI and optimization methods based on MIP formulation. We implemented a DSS to incorporate both AI techniques and OR methods. The experiments in simulated environment provided promising scheduling results. In the future, we may extend the DSS and refine both the planning method and optimization solution methods to improve further the performance of the scheduling solutions.

Acknowledgement This research is supported in part by University of Macao Research Grant under grant number RG047/01-02S /WGH/FBA.

References